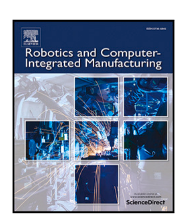
Robotics and Computer–Integrated Manufacturing 93 (2025) 102918

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim



Full length article

Integrated optimization of storage space allocation and crane scheduling in automated storage and retrieval systems

Wenbin Zhang [a,b], Zhiyun Deng [a], Chunjiang Zhang [a], Weiming Shen [a,b,*]

[a] *School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, 430074, Hubei, China*
[b] *School of Mechanical and Electrical Engineering, Kunming University, Kunming, 650214, Yunnan, China*

ARTICLE INFO

Keywords:
Automated storage and retrieval system
Integrated optimization
Multi-layer adaptive length coding
Energy efficiency
Hybrid genetic algorithm

ABSTRACT

This paper addresses the challenge of integrated optimization for storage space allocation and crane scheduling in automated storage and retrieval systems. The problem encompasses tasks such as assigning storage/retrieval requests, allocating storage spaces, and planning crane routes within each operation cycle. To tackle this, we introduce a multi-layer adaptive length coding method to effectively map the solution space to the problem space. Employing a coevolutionary framework, we decompose and process the integrated optimization problem, further optimize it with a hybrid genetic algorithm. Numerical experiments across a wide range of scenarios are conducted to evaluate the algorithm's performance under varying request sizes and crane capacities. The introduction of the coevolutionary framework improves optimization by up to 14.78%, with an average improvement of 34.09% compared to the method currently used in the company. In addition, we introduce a novel optimization metric, termed potential energy consumption, designed to enhance system energy efficiency. Comparative analysis against metrics like makespan reveals the superiority of our proposed approach in terms of coverage and optimality, particularly in large-scale scenarios. The combined implementation of integrated optimization and the new evaluation metric leads to substantial energy cost savings for real-world automated storage and retrieval systems.

1. Introduction

Automated Storage and Retrieval System (AS/RS) epitomizes the seamless integration of modern warehousing, computer, and automation technologies [1]. Equipped with a diverse array of automated logistics devices [2], AS/RS ensures swift, precise, and efficient material handling around the clock [3].

The crane is an automated transport equipment that navigates within the aisles of the AS/RS, traveling between the cargo spaces and the input/output (I/O) port, as illustrated in the upper half of Fig. 1.

Multi-shuttle cranes are capable of carrying multiple Stock Keeping Units (SKUs) in parallel, with the number of shuttles equaling the crane's capacity [4]. This advancement has notably boosted the throughput of AS/RS.

The period from when the multi-shuttle crane departs the I/O port until its return is termed an operation cycle [5]. Within a single cycle, the crane can fulfill multiple storage and retrieval (R/S) requests, also referred to as dual-command cycles [6]. The task pool keeps a record of all R/S requests received by the AS/RS within a specific timeframe. The crane requires several operation cycles to complete them [3]. Therefore, the primary challenge lies in devising an optimal method to allocate the R/S requests in the task pool to each operation cycle of crane.

Within each operation cycle, the crane departs from the I/O port, transporting items designated for storage. It first targets an empty space (E) on the shelf to deposit an SKU. Subsequently, the crane can either proceed to another E space for a similar operation or move to a retrieval space (R) to fetch an SKU. This process continues until the number of storage and retrieval requests equals the number of loads, a challenge known as the Multi-Shuttle Crane Scheduling Problem (MSCS) [7]. Notably, after retrieving an SKU from an R space, the crane has the capability to deposit another SKU in the same space referred to as RS operation. This practice, known as shared storage, reduces one movement within an operation cycle [7,8] and creates a Shared Storage Multi-Shuttle Crane Scheduling Problem (SSMSCS) [9]. However, this approach limits flexibility in storage space allocation and narrows the solution space [10]. Hence, formulating a well-considered scheduling plan for the storage (S), retrieval (R), and RS operations within each operation cycle stands as another crucial consideration.

* Corresponding author at: School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, 430074, Hubei, China.
E-mail address: wshen@ieee.org (W. Shen).

https://doi.org/10.1016/j.rcim.2024.102918
Received 11 December 2023; Received in revised form 21 November 2024; Accepted 22 November 2024
Available online 5 December 2024
0736-5845/

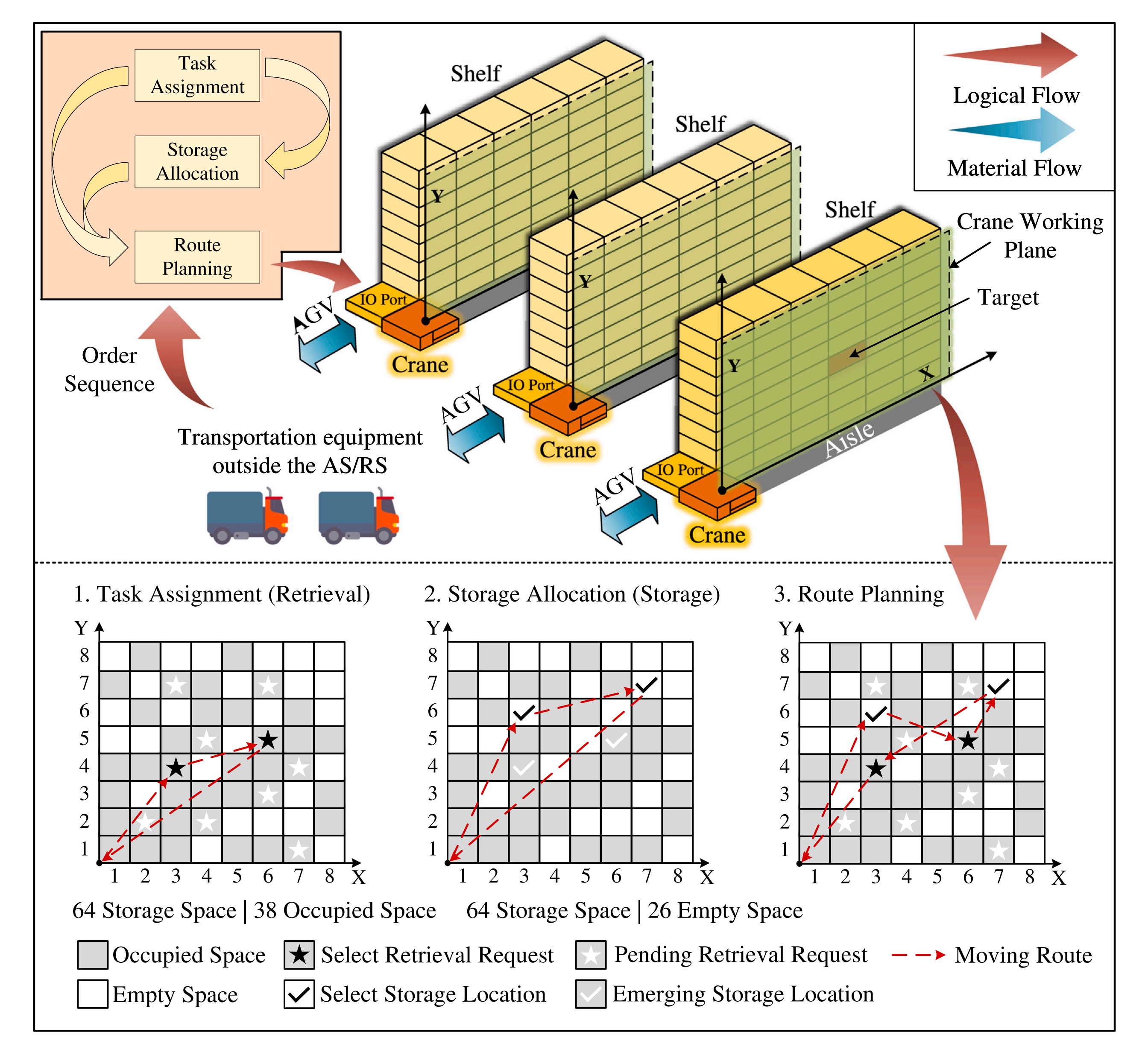


Fig. 1. Illustration of the workflow of the automatic storage/retrieval system.

Moreover, the operational energy costs of logistics systems tend to surpass construction expenses within a few years [11,12]. Surprisingly, this facet has received relatively little attention from scholars investigating crane scheduling issues in AS/RS. Hence, we introduce a novel optimization metric termed Potential Energy Consumption (PEC) and provide a clear calculation methodology. This proposed metric offers a more comprehensive perspective and yield long-term benefits to AS/RS. It can also determine the optimal number of shuttles for construction design, considering variables such as target throughput, shelf utilization, and more.

In summary, this endeavor encompasses the following key challenges, as illustrated in the lower half of Fig. 1: *(a) Task Assignment*: A rational approach is required for assigning R/S requests from the task pool to each operational cycle of the crane. *(b) Storage Space Allocation*: Addressing the reduction in flexibility in storage space allocation when employing shared storage. *(c) Crane Route Planning*: The unpredictability in the number of movements renders traditional crane scheduling methodologies outdated, necessitating a fresh approach to solve the crane route planning problem. This undertaking presents a highly practical yet intricate and demanding task.

However, this paper makes the following contributions to tackle these challenges: *(a) Integrated Optimization*: We integrate storage space allocation with crane scheduling, thereby transforming RS movement into a versatile component and granting cranes with expanded operational possibilities. *(b) Novel Coding Method*: A multi-layer adaptive length coding is introduced to enable the mapping from solution space to problem space. The adaptability of code length entirely addresses the issue posed by the varying number of movements during each operation cycle, rendering the problem highly scalable. *(c) Co-evolutionary Algorithm*: A new heuristic algorithm named HGA-CC is proposed, which demonstrates remarkable search efficiency in the context of integrated optimization with structurally similar subproblems.

The remainder of the paper is structured as follows: Section 2 discusses the related work on automated storage and retrieval systems. The mathematical formulation of the integrated optimization is given in Section 3, followed by a detailed description of the multi-layer adaptive length coding method and the algorithm principles in Section 4. Numerical simulation experiments are conducted in Section 5 to assess the proposed algorithm and the optimization metric. Section 6 concludes this paper and discusses avenues for future research.

2. Related work

In recent decades, the demand for large-scale logistics has propelled the advancement of automated storage and retrieval systems [13]. The integration of Internet of Things (IoT) has facilitated the seamless automation of material storage, retrieval, conveying, and sorting, substantially elevated logistics efficiency [14,15].

AS/RS commonly incorporate various automation equipment, including automated guided vehicles (AGVs) [16], rail-guided vehicles (RGVs) [17,18], conveyors [2], and cranes [19]. Scholars have delved extensively into the scheduling of these devices to enhance overall operational efficiency.

2.1. R/S requests assignment and storage allocation

AS/RS requests arrive in batches [20] and are subsequently stored in a task pool. The First Come, First-Served (FCFS) approach is the simplest for assigning request sequences and is widely adopted [6]. However, this method is inefficient and offers significant potential for optimization [21]. Bozer et al. [22] proposed a Nearest Neighbor (NN) to optimize the request sequence and enhance the throughput capacity of the system.

To designate suitable storage spaces for SKUs, a storage allocation strategy is essential. Francis et al. [23] outlined four primary allocation approaches and demonstrated that shared storage offers the most efficient space utilization. Sui et al. [24] developed a multi-objective optimization model considering factors like cargo turnover, shelf center of gravity stability, and cargo correlation, using categorized storage. They solved this model with a genetic algorithm. Zhao et al. [25] employed shared storage, incorporating SKU quality and processing order information to determine optimal storage spaces through an ant colony algorithm. The ABC storage strategy is also widely adopted [26,27]. It proves advantageous when storage requests arrive randomly but is more subjective. However, limitations arise when allocating storage spaces without considering the crane's route planning.

2.2. Crane operation scheduling

Scheduling crane operations, which involves planning the movement routes of cranes during operation cycles, significantly influences both throughput and efficiency of AS/RS [28]. Yang et al. [9] conducted research on the multi-shuttle crane scheduling problem, considering $2-n$ storage/retrieval spaces. They formulated an integer programming model and solved it with heuristics. Their findings highlighted that the shared storage strategy can significantly reduce the makespan. Nonetheless, this approach limits flexibility in storage space allocation.

Some scholars have also transformed the crane scheduling problem into a vehicle routing problem (VRP) or capacitated vehicle routing problem (CVRP) for solving. Polten et al. [7] addressed a scenario akin to Yang et al. [9], involving shared storage and multi-shuttle cranes. They reformulated the problem as a specialized form of CVRP and precisely solved it with a well-established vehicle routing toolbox. Boysen et al. [29] focused on a single-capacity crane scenario and discussed the impact of buffer sizes on throughput. They demonstrated that the crane scheduling problem is strongly NP-hard and devised exact solutions using a VRP solver. However, their methodologies are tailored for addressing crane scheduling scenarios with a fixed number of movements and are not suitable for handling more flexible situations with an uncertain number of movements.

In summary, as far as we know, current research approaches still fall short in integrated optimizing task assignment, storage allocation, and crane route planning in AS/RS. This gap in the literature has motivated us to undertake this study.

3. Problem definition

Consider the specific scenario depicted in Fig. 1 to elucidate the system's workflow. In this scenario, the AS/RS encounters a set of N storage and N retrieval requests in the task pool. The crane's capacity (expressed as the number of shuttles) is denoted by M. Notably, all tasks assigned to the crane are dual-command. This means that M storage and M retrieval requests from the task pool can be completed during one command cycle. To fulfill all tasks, the crane returns to the I/O port for goods exchange a total of T times, where $T=N/M$. For convenience, we assume that N is an integer multiple of M. This assumption can be easily accommodated by introducing pseudo-tasks. As an example, a feasible combination of request numbers, shuttle quantity, and crane trips could be: $N=30$, $M=5$, and $T=6$.

To enhance operational flexibility, crane movements are categorized into three types: R movements, S movements, and RS movements. During an R movement, the crane executes a retrieval task upon reaching the designated destination. Conversely, an S movement involves a storage request after the crane reaches its destination. Additionally, an RS movement includes a SKU exchange, where the crane unloads one SKU and replaces it with another. The introduction of RS movements provides the crane with additional options, as illustrated in Fig. 2.

The inclusion of RS movement introduces variability in the total number of movements across different operation cycles, with the maximum number of movements satisfying $j_{\max}\in[M+2,2M+1]$. We denote the current operation cycle as $i\in 1,2,\dots,T$, and represent the individual movements within the operation cycle as $j\in\{1,2,\dots,j_{\max i}\}$. Notably, when the crane is equipped with only one shuttle, the crane operation cycles revert to a fixed pattern of $IO\rightarrow S\rightarrow R\rightarrow IO$, where $j_{\max}=3$.

Finally, according to the practical situation of the system, the following assumptions are introduced to establish the mathematical model for this problem in a more concise manner:

(a) The locations of all retrieval requests are known in advance.

(b) The locations of all empty cargo space are pre-determined.

(c) At the initiation of the first task, the number of empty shelves surpasses the quantity of incoming goods, necessarily.

(d) For the sake of research, the energy consumption resulting from the shuttle's access to goods after reaching the target location is not taken into account.

(e) The frequency of accessing goods remains uniform over a given period of time.

These assumptions provide a structured framework for the mathematical model and help streamline the system analysis.

3.1. Decision variables

Without loss of generality, we set the position of the I/O port as the origin. Then, in the jth move of the ith operation cycle, we denote the column and tier coordinates of the target as x_{ij} and y_{ij}, respectively. The relative displacements in the X and Y directions produced by this move are:

$$dX_{ij}=\begin{cases}|x_{i,j}-x_{i,j+1}|,\ j\in\{1,2,\dots,j_{max}-1\}\\|x_{i,j}|,\ j=j_{max}\end{cases}\quad\forall i,\tag{1}$$

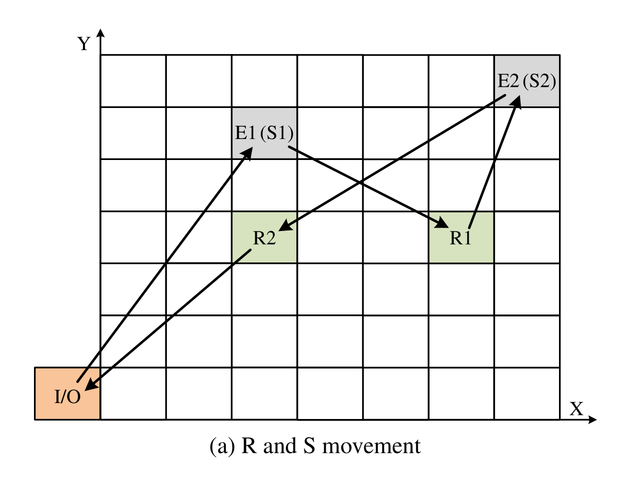
$$dY_{ij}=\begin{cases}|y_{i,j}-y_{i,j+1}|,\ j\in\{1,2,\dots,j_{max}-1\}\\|y_{i,j}|,\ j=j_{max}\end{cases}\quad\forall i.\tag{2}$$

The following decision variables for determining the type of crane movement attribute in the operation cycle are defined for providing a clearer description of the storage and retrieval process:

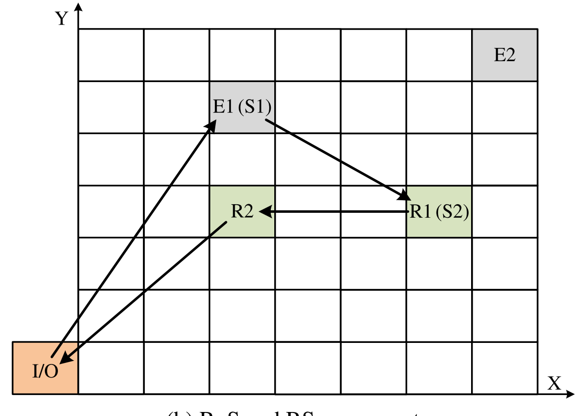
$z^R_{i,j}$: 0-1 variable, $z^R_{i,j}=1$ if the jth move of the ith operation cycle is performed as a retrieval movement; otherwise $z^S_{i,j}=0$.

$z^S_{i,j}$: 0-1 variable, $z^S_{i,j}=1$ if the jth move of the ith operation cycle is performed as a storage movement; otherwise $z^S_{i,j}=0$.

$z^{RS}_{i,j}$: 0-1 variable, $z^{RS}_{i,j}=1$ If the jth move of the ith operation cycle is performed as a RS movement; otherwise $z^{RS}_{i,j}=0$.



(a) R and S movement



(b) R, S and RS movement

Fig. 2. Changes of crane operating method after adding the RS movement.

$z^V_{i,j}$: 0-1 variable, $z^V_{i,j} = 1$ If the jth move of the ith operation cycle is valid; otherwise $z^V_{i,j} = 0$. In particular, the roles of $z^V_{i,j}$ and $j_{\max}$ are similar and are reflected in the implementation of variable coding lengths. $j_{\max}$ can be derived from Eq. (3), while $z^V_{i,j}$ can be easily derived from $j_{\max}$ by Eq. (4).

$$j_{\max i} = \min \kappa,\ if \sum_{j=1}^{\kappa} (z^R_{i,k} + z^{RS}_{i,k}) \geq M, \tag{3}$$

$$z^V_{i,j} = \begin{cases} 0, & if\ j > j_{\max} \\ 1, & otherwise \end{cases} \quad \forall i,\ j. \tag{4}$$

In Eq. (4), it is obvious that $j_{\max}$ is the maximum index of the non-zero value of $z^V_{i,j}$, which is frequently used in the calculation of the equations in this section.

3.2. Constraints

3.2.1. Shuttle load constraints

At the outset of each task, the crane is loaded with storage goods. Hence, the initial load of the crane, denoted as $m_{i,1}$, adheres to the following conditions:

$$m_{i,j} = M, \qquad \forall i, \forall j = 1, \tag{5}$$

$$m_{i,j} - m_{i,j-1} = (z^R_{i,j} - z^S_{i,j}) z^V_{i,j}, \forall i, \forall j \in \{2, \ldots, j_{\max}\}, \tag{6}$$

$$m_{i,j} \leq M, \qquad \forall i, j. \tag{7}$$

Here, M represents the number of shuttles. As subsequent movements unfold, the quantity of goods carried by the crane, denoted as m_{ij}, can be iteratively calculated in accordance with Eqs. ((5),(6)).

Constraint (7) specifically mandates that, during the execution of any request, the quantity of goods $m_{i,j}$ carried must not exceed the capacity of the crane.

$$\prod_{i=1}^{T} z^S_{i,1}(1 - z^R_{i,1})(1 - z^{RS}_{i,1}) = 1, \forall i. \tag{8}$$

According to Eqs. (5)–(7), due to being fully loaded, the crane will invariably commence by visiting an available space and depositing a SKU. Hence, the initial request of each cycle must be constrained as a storage request, as represented in Eq. (8).

3.2.2. Total number of orders constraint

Given the context of the problem, the combined total of storage and retrieval requests in a batch of orders is denoted as N. Thus, we formulate it as follows:

$$\sum_{i=1}^{T} \sum_{j=1}^{j_{\max}} z^R_{i,j} + z^{RS}_{i,j} = N, \tag{9}$$

$$\sum_{i=1}^{T} \sum_{j=1}^{j_{\max}} z^S_{i,j} + z^{RS}_{i,j} = N, \tag{10}$$

$$z^R_{i,j} + z^S_{i,j} + z^{RS}_{i,j} = 1, \forall i, j. \tag{11}$$

Eq. (9) indicates that the total number of retrieval requests amounts to N. Eq. (10) means that the total number of storage requests also equals N. Furthermore, Eq. (11) shows that each move must be one of three categories: R, S or RS movement.

3.2.3. Shelf space constraints

This constraint stipulates that within a single cycle, revisiting the same location is prohibited, as it contradicts the operational requirements of the crane, Denoted by constraints (12):

$$x_{i,j} \neq x_{i,j'}\ or\ y_{i,j} \neq y_{i,j'}, \forall j, j' \in \{1, 2, \ldots, j_{\max}\}, j \neq j', \tag{12}$$

$$1 \leq x_{i,j} \leq L_x, \tag{13}$$

$$1 \leq y_{i,j} \leq L_y. \tag{14}$$

Meanwhile, constraints (13), (14) establish that the movement range of the crane must be confined within the aisle and should not extend beyond the boundaries. Where L_x and L_y represent the depth and height of the warehouse aisle, respectively.

3.3. Objective function

The crane's movement in the X and Y axes relies on horizontal and vertical motors [30]. Our energy-centric objective function is intricately designed down to the motor level and comprises two key components: The first component encompasses the aggregate energy consumed by the horizontal and vertical motors during the completion of all requests over T operation cycles. The second component involves the product of the throughput frequency and the retrieval energy consumption of storage requests, representing the predicted retrieval energy consumption for stored SKUs over a specific period. These two components are mathematically formalized in Eqs. (15), (16), respectively.

$$f_1(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{T} \sum_{j=1}^{j_{\max}} F(dX_{i,j}, dY_{i,j}, Q_{i,j}), \tag{15}$$

$$f_2(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{T} \sum_{j=1}^{j_{\max}} F(x_{i,j}, y_{i,j}, q_{i,j}) \cdot (z^S_{i,j} + z^{RS}_{i,j}) \cdot \chi_{i,j}. \tag{16}$$

Here, F denotes the energy consumption formula. It encompasses the energy expended by the horizontal and vertical motors during the crane's acceleration, deceleration, and uniform-speed movement along the X and Y axes. $Q_{i,j}$ represent crane load in the jth movement of the ith task. $q_{i,j}$ represent quality of SKU handled in the jth movement of the ith task. The second component of the objective function pertains

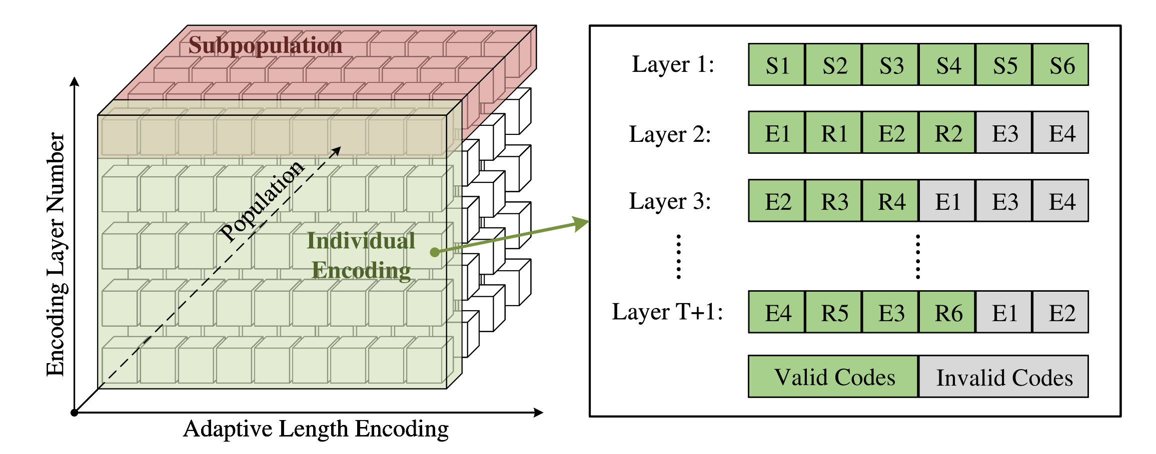


Fig. 3. Coding populations and an example of coding.

to the estimated retrieval energy consumption of storage requests. This aspect offers flexibility in selecting storage spaces while optimizing the crane route.

Hence, the two components are combined to derive the overall objective function, referred to as potential energy consumption (PEC), which can be neatly expressed as:

$$\min_{\boldsymbol{x},\boldsymbol{y}}\{f_1(\boldsymbol{x},\boldsymbol{y})+f_2(\boldsymbol{x},\boldsymbol{y})\}. \tag{17}$$

Here, χ is the throughput frequency of goods over a statistical period of time. To illustrate, for instance, let us assume cargo A has accumulated 528 records of storage and retrieval movements within the past two months (60 days). Applying our method, we calculate the frequency of this cargo as:

$\chi = 528/60 = 8.8.$

Similarly, cargo B with only 12 storage and retrieval, yields a frequency of 0.2. This clearly indicates that cargo A experiences a significantly higher throughput frequency compared to cargo B. Consequently, it becomes evident that we should position cargo A on the lower shelf, in proximity to the I/O port. This philosophy anticipates its future high-frequency retrieval and aligns with a forward-looking approach.

4. Integrated optimization for storage spaces and crane scheduling with co-evolution

The problem addressed in this article can be classified as Problem[F, k|IO^2, open|*] according to Boysen et al. [6]. Here, F denotes a single IO-point positioned at the front end of the rack. k signifies the use of multiple shuttles for a single crane. IO^2 refers to the implementation of a dual-command loop. The term "open" indicates that storage requests allow goods to be placed on any available open shelf. The asterisk denotes that the objective function cannot be explicitly categorized.

4.1. Solution representation

In formulating the solution, it is crucial to acknowledge that the crane executes a variable number of movements in operation cycles. This variability in solving subproblems related to crane route planning led to the introduction of a Multi-layer Adaptive Length Coding (MALC) to accommodate this characteristic.

An illustrative solution for a problem with $N = 6$, $M = 2$, and $E = 4$ is depicted in Fig. 3. In this representation, the first layer of coding consists of a sequence with a length of N. Each element, denoted as $S_1 - S_i$ where i ranges from 1 to N, corresponds to the N storage requests. These requests are loaded onto the crane in the order specified by the coding, with the crane capable of handling up to M storage requests simultaneously. Similarly, the coding for layers $2 \sim T+1$ comprises several sequences of equal length. Each of these sequences signifies the movement information of complete operation cycles. This results in a total of T sequences. The length of each sequence is determined by $M+E$, where E denotes the number of vacancies.

It is specified that the last R-code, along with its preceding sequence, constitutes a valid code in each sequence, while the remaining portion is deemed invalid. Each valid code represents a movement of the crane. An E signifies the execution of a storage movement, while an R indicates the execution of a retrieval or an RS movement, which will be determined based on the subsequent calculation. The effective encoding length is denoted as $J_{\max}$, inherently satisfying $M+2 \le J_{\max} \le 2M+1$. The maximum length of the effective encoding occurs when there are no RS requests, resulting in $J_{\max} = 2M+1$. Building upon this, for each additional RS request, the code length is reduced by one, until only one R task remains (as the last task must be of type R), yielding a minimum coding length of $M+2$.

4.2. Decoding method

4.2.1. Judgment of RS requests

Decoding serves as the inverse operation of encoding. it is crucial to determine the crane's movement route and precisely identify the RS requests within a coding. The parsing process as depicted in Fig. 4, first, we eliminate the invalid code segment. Next, we scan from left to right to locate E requests. Upon finding them, we search for corresponding R requests under the condition that the subsequent request following this R cannot also be an R. The matched E and R are then removed from the coding, until all E requests have been cleared, leaving only the remaining RS requests.

Notably, if an E request exists in the remaining code, it indicates an overload of the crane during task execution. Hence, this is an invalid sequence to be corrected. This parsing process for RS requests can be represented in pseudo-code as outlined in Algorithm 1.

4.2.2. Combined decoding of multi-layer coding

When decoding a feasible solution, it is crucial to integrate the storage requests from the first layer encoding with the route details and retrieval requests from the $2 \sim T+1$ layers. This process entails sequentially aligning all E and RS requests from the $2 \sim T+1$ layers encoding with the corresponding S requests from the first layer. For example, as illustrated in Fig. 5, Position $E1$ corresponds to $S1$, and similarly, position $E2$ corresponds to $S2$. As $R3$ is an RS request, it also corresponds to $S4$.

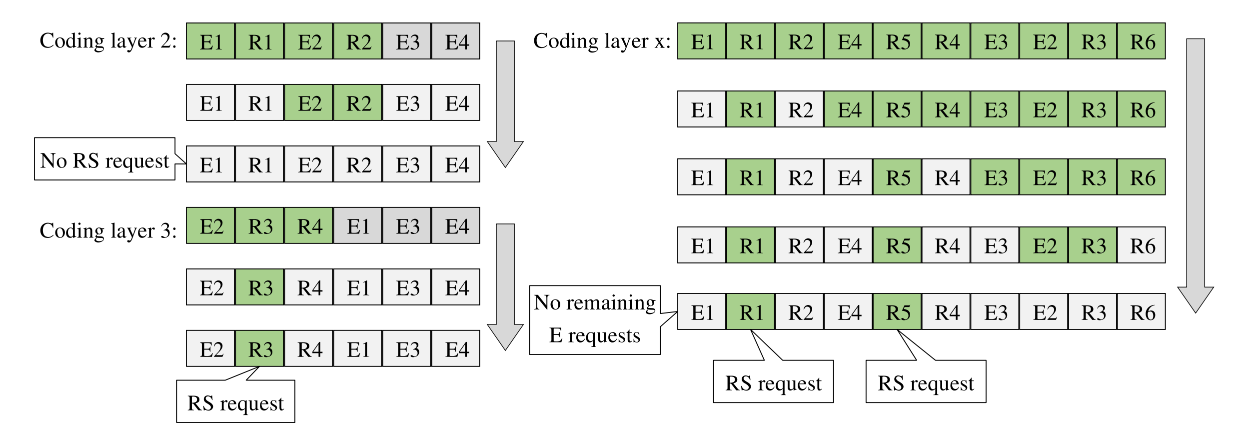


Fig. 4. Parsing process for RS request. left: Parsing process in the instance above, right: Parsing process for a more complex instance.

Algorithm 1: Parsing *RS* requests

```
Data: Code
X ← getValidParts(Code);
while E is found in X do
    foreach i ← find(E) + 1 to length(X) do
        if i==length(X) then
            X ← X.remove(E.index)
            X ← X.remove(i)
            break
        else
            if X(i) is R and X(i + 1) is not R then
                X ← X.remove(E.index)
                X ← X.remove(R.index)
                break
            else
                continue
            end
        end
    end
end
R/S_List← X
Result: R/S_List
```

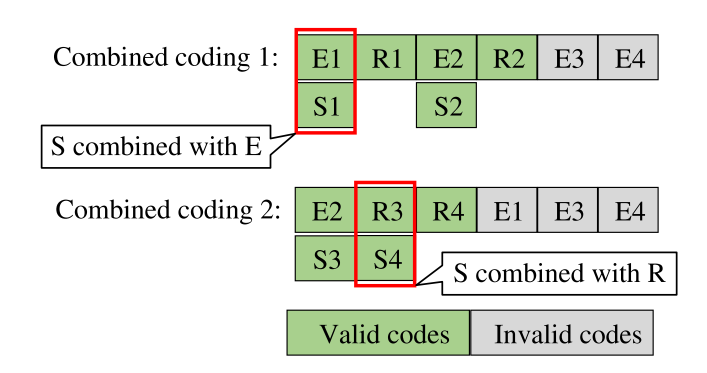


Fig. 5. Combinatorial decoding process for multi-layer adaptive length coding.

4.2.3. Update of vacant space record table

As the vacant spaces on the shelves undergo constant changes during the crane's operations, while the numbering of these spaces in the code remains static, we need to devise a method to update these numbers. This process is illustrated in Fig. 6. In route 1, the crane first moves to position $E1$ to store a SKU, then proceeds to position $R1$ for retrieval. At this point, the empty space shifts from the $E1$ position to the $R1$ position. This transition is recorded in the vacant space record table as the key–value pair $E1 - R1$. The same applies to $E2 - R2$. In route 2, the crane moves to position $E2$, which corresponds to the location of $R2$ in the vacant space record table. The second stop at $R3$ falls under the RS category, causing no change in the vacant space. Finally, the crane heads to $R4$, leading to another shift in the vacant space, this time to the position of $R4$. Consequently, $E2-R2$ is updated to $E2 - R4$ in the vacant space record table. The crane then returns to the I/O port, concluding the work cycle.

4.3. Co-evolutionary framework

Given that the integrated problem encompasses a multitude of sub-problems, including task assignment, storage allocation, and route planning, most of which are classified as NP-Hard, the solution space for the overall problem is exceedingly vast. In this section, we introduce a novel co-evolutionary algorithm to efficiently address such a complex multi-level optimization challenge. Our algorithm draws inspiration from Cooperative Co-evolutionary (CC) principles and leverages Hybrid Genetic Algorithm (HGA), employing a partitioning strategy to tackle large-scale and intricate multi-level optimization problems [31]. This approach is particularly effective when optimizing problems involve subproblems that exhibit high similarity following decomposition, allowing for faster search efficiency within constrained timeframes.

As illustrated in Fig. 7, our proposed co-evolutionary framework comprises three core components: problem Decomposition, subcomponent optimization, and global optimization.

4.3.1. Problem decomposition

In this section, based on the T combined encodings depicted in Fig. 5, we dissect the overarching problem, which involves T moving cycles of the crane, into T sub-problems. Each sub-problem represents a distinct operation cycle, resembling a specialized VRP/CVRP route planning scenario. Consequently, we utilize state-of-the-art evolutionary operators specifically tailored to address these decomposed sub-problems.

Similarly, a local fitness function is defined in the ith sub-problem expressed in Eqs. (18)–(20) as follow:

$$g_1(\boldsymbol{x}, \boldsymbol{y}) = \sum_{j=1}^{j_{\max}} F(dX_{i,j}, dY_{i,j}, Q_{i,j}), \ \forall i, \tag{18}$$

$$g_2(\boldsymbol{x}, \boldsymbol{y}) = \sum_{j=1}^{j_{\max}} F(x_{i,j}, y_{i,j}, q_{i,j}) \cdot (z^{S}_{i,j}, z^{RS}_{i,j})\chi_{i,j}, \ \forall i, \tag{19}$$

$$\min_{\boldsymbol{x},\boldsymbol{y}} \{g_1(\boldsymbol{x}, \boldsymbol{y}) + g_2(\boldsymbol{x}, \boldsymbol{y})\}, \ \forall i. \tag{20}$$

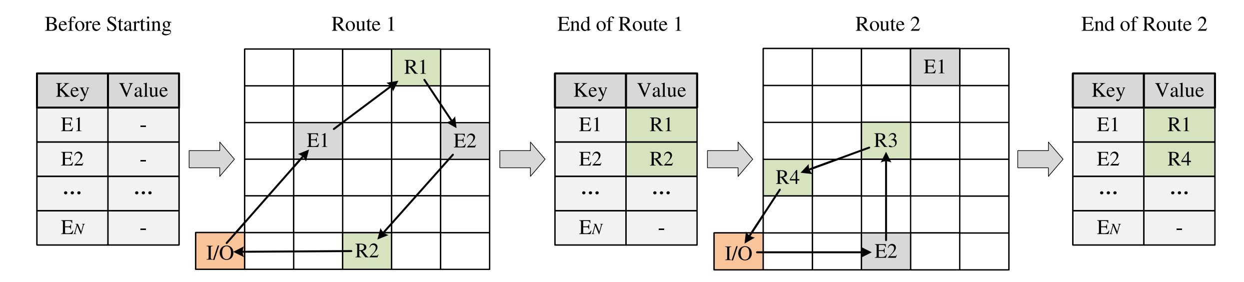


Fig. 6. Updating information of empty cargo space according to the vacancy record table.

Fig. 7. Flow chart of Hybrid Genetic Algorithm with Cooperative Co-evolution.

The local fitness function shares a striking resemblance with the global fitness function in both representation and symbolic definition, and will not be reiterated here. Additionally, as illustrated in Fig. 3, we amalgamate single-operation cyclic subproblems from the same layer across different populations to create T subpopulations, enabling subpopulation optimization.

4.3.2. Subpopulation optimization

We devised the evolution operator based on an approach used to address the TSP/VRP problem [32]. However, since our route encoding incorporates R, S, and E types, a direct application is not straightforward. It is imperative to uphold the constraints outlined in Section 3.2. when employing the evolution operator.

For this reason, we refrain from modifying the evolution operator to explicitly accommodate our constraints. Instead, we focus on rectifying newly generated infeasible solutions without altering the evolution operator itself.

In accordance with the Load Capacity Constraints outlined in Section 3.2, multiple invalid codes may arise that violate the constraint. These invalid codes can be rectified using the Two-Step Repair method: In the first step, all invalid codes that do not commence with an E are identified. Then search in reverse for the first occurrence of E and insert it into the head of the code. In the second step, once an E emerges without any remaining items on the crane, it is inserted into the tail of the code, as depicted in Fig. 8.

These two straightforward steps empower us to seamlessly employ a diverse range of evolutionary operators, adapted from the TSP/VRP problem on our code. This facilitates efficient evolutionary operations on subpopulations without concerns about generating infeasible solutions resulting from the application of evolutionary operators.

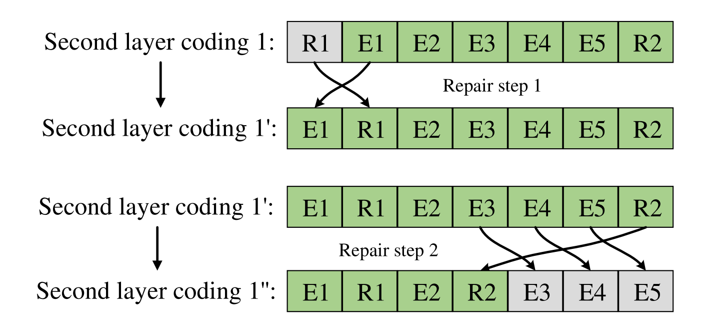


Fig. 8. The two-step restoration process for an invalid coding.

4.3.3. Global optimization

During the global optimization phase, we implemented a Hybrid Genetic Algorithm tailored to the requirements of integrated optimization. Acknowledging the potentially disruptive impact of the crossover on multi-layer coding, we have chosen a local neighborhood search as an alternative. The global optimization process includes a selective mutation and a local search.

(a) Selective Mutation: In this phase, we undertake a dual approach. Firstly, for the first-layer encoding (S encoding), we employ a roulette wheel selection to pick 1 elite individual from every n individuals. Subsequently, these elite individuals undergo mutation through the application of two-point crossover, head insertion, and interval inversion operators. Secondly, for the coding of layers $2 \sim T+1$, a similar procedure ensues. Elite individuals are chosen, and all extracted R coding from the multitude of route sequences undergo mutation. These mutated codes are then reinserted into their original sequence positions, thereby yielding a new generation of mutant individuals.

(b) Local Search: During the local search, our attention turns to the $2 \sim T+1$ layers encoding only. Within this layer, we randomly choose a set of sequences. Here, we modify the positions of the E codings in these selected sequences. Subsequently, we apply the Two-Step Repair outlined in the preceding section to rectify any infeasible solutions that disobedience to constraints.

5. Numerical simulation experiments

In this section, computational experiments are conducted to evaluate the performance of the proposed algorithm HGA-CC. We compare it with three other approaches: the basic Hybrid Genetic Algorithm (HGA), the classical Genetic Algorithm (GA), and the Random Evolution (RE). The HGA performs global optimization only, without using the concept of co-evolution. The GA retains the original crossover strategy and does not employ local search. Meanwhile, the RE is closer to the First Come, First Served (FCFS) method used in the company. We generated a range of test instances, spanning from small to large scenarios. These instances encompass diverse physical sizes of AS/RS, varying numbers of shuttles, and different request quantities. The physical parameters are detailed in Table 1.

All code in this section was developed using MATLAB 2023a on a PC with a 13th Gen Intel(R) Core(TM) i5-13500H processor (2.60 GHz), 16 GB RAM, and the Windows 11 operating system. The algorithm parameters were fine-tuned using the Design of Experiment (DOE). Finally, we conduct a thorough analysis of the results to assess the performance of the proposed algorithm and the effectiveness of the evolutionary.

5.1. Scenario generation

To evaluate scenarios at different scales, we developed an instance generation program. This program generates AS/RS specifications, including R/S request details as well as shelf specifics, based on the defined parameters such as the number of tasks T and the initial count of empty spaces E.

Table 1
Numerical experimental parameters.

Description	Notation and value
Population	50
Maximum number of iterations	500
Number of experiments	n = 20
Storage/retrieval frequency	$\chi \in [1, 600]$
Load (kg)	$q \in [1, 20]$
Crane weight (kg)	$G = 160$
Wheel diameter (mm)	$D = 145$
Gravitational acceleration (m/s^2)	$g = 9.8$
Rolling friction arm of the wheels along the track (mm)	$K = 0.5$
Bearing friction coefficient	$U = 0.02$
Bearing inner diameter (mm)	$d = 20$
Friction coefficient	$f = 0.03$
Slope resistance coefficient	$K_r = 0.001$
Wind resistance coefficient (ignored indoors)	$K_w = 0$
Horizontal/vertical speed (m/s)	$v_s/v_r = 1$
Acceleration (m/s^2)	$\alpha = 0.5$
Institutional operational efficiency	$\eta = 0.9$

We choose a cubicle shelf design, as shelves with the "square in time" feature tend to offer the highest storage efficiency [15]. Additionally, we assume an initial shelf utilization rate of 50%. Since the order pool is initialized with a number of SKUs already on the shelves, retrieval orders involve only a fraction of them. The number of tiers and columns of initialized shelves can be expressed using Eq. (21) as:

$$L_x = L_y = \lceil \sqrt{\frac{N+E}{\chi}} \rceil \tag{21}$$

In addition, the capacity range of the cranes was set between 2 – 8 to test the effectiveness of the proposed approach in handling more complex and larger instances. Additionally, we assign attributes such as weight and throughput frequency to R/S requests in accordance with the limits specified in Table 1.

5.2. Comparison results of different algorithms

In this section, we compare the performance of the four algorithms on instances of various sizes. Each algorithm and instance undergoes 20 independent tests. The optimization results based on the fitness function are presented in Table 2. Here, "Group" indicates the instance number, and $N \times M \times T$ represents the combined parameters of total requests, shuttle count, and operation cycle number for different instances. The variance of the fitness function and the mean running time are also presented in Table 3. Additionally, we introduce the GAP as a metric to quantify the efficiency between algorithms, defined as follows:

$$GAP_i = \frac{|f_i - f_{LB}|}{f_{LB}} \tag{22}$$

Here, f_{LB} denotes the lower bound (optimal value) of the obtained solution. GAP serves as an indicator of how closely the algorithm being compared approaches the best. The variance of the fitness, represented as s_2, is defined as follows:

$$s_2 = \frac{\sum_{i=1}^{n}(f_i - \bar{f})^2}{n} \tag{23}$$

Here, $\bar{f}$ stands for the average of the experimental results, and n signifies the total number of experiments.

From Tables 2 and 3 can observe that among them: RE exhibits the fastest execution speed, yet it yields the least optimal results. HGA outperforms GA, with both exhibiting similar speeds, although GA shows less stability. The proposed HGA-CC not only achieves the optimal solution consistently in each instance, but also searches faster compared

Table 2
Analysis of numerical experimental results.

Group	N × M × T	Average fitness			
		RE	GA	HGA	HGA-CC
1	10 × 2 × 5	29.79582	23.20681	22.43690	18.79328
2	16 × 4 × 4	49.66966	38.35205	37.52791	31.18683
3	30 × 6 × 5	139.1435	110.9490	106.5227	90.47085
4	50 × 5 × 10	221.9264	175.1614	165.9692	144.2095
5	80 × 4 × 20	478.2021	383.9444	374.8291	322.2231
6	100 × 4 × 25	561.1354	455.2398	444.1340	380.8032
7	120 × 3 × 40	800.2101	658.4731	638.2395	544.1779
8	150 × 5 × 30	1001.184	817.9538	792.5851	682.5047

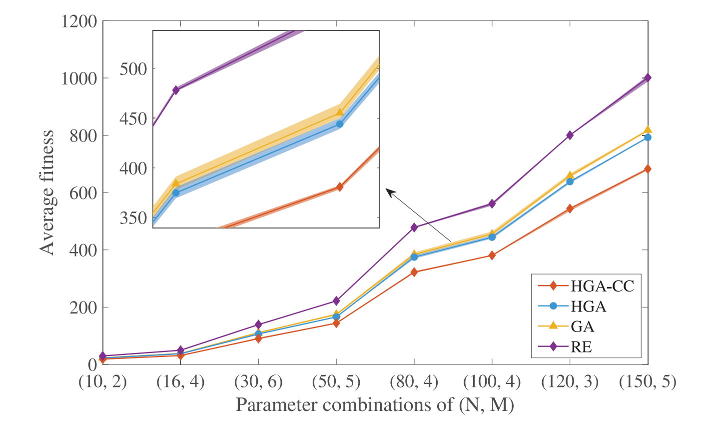


Fig. 9. Optimization results of the algorithms for different instances.

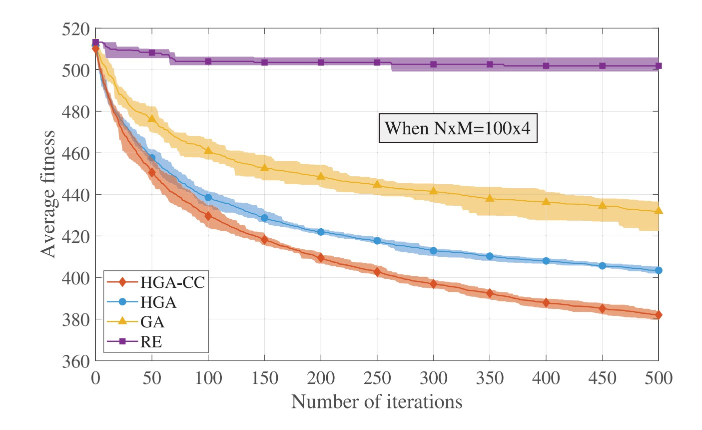


Fig. 10. Iterative process of different algorithms.

to HGA and GA due to the adept optimization of the subproblems using a co-evolutionary. In this context, the average GAP between HGA-CC and the second-best-performing HGA reached 14.78%.

To facilitate a clearer comparison of algorithm performance across various instance sizes, we have visualized the experimental results. In Fig. 9, the range of optimal value obtained by the four algorithms is presented, sorted in ascending order based on the number of tasks N.

Furthermore, the iteration curves of the different algorithms are plotted for an instance of size $N \times M = 100 \times 4$, as shown in Fig. 10. The shaded areas indicate the upper and lower bounds identified by the algorithms, while the solid line shows the average. It can be observed that among these algorithms, RE exhibits the lowest optimization efficiency, indicated by the slowest rate of curve descent. HGA demonstrates higher search efficiency than GA and shows more stable search results, as evidenced by the smaller shaded area. The proposed HGA-CC displays the highest search efficiency and steadily converges towards the best solution.

To provide a comprehensive view of the fulfillment process, A simulation of storage and retrieval within each run cycle was conducted In Fig. 11. It is four complete operation cycles (Route1~Route4) for an instance of size $N \times M \times T = 12 \times 3 \times 4$. This visualization demonstrates the crane's execution of R/S operations at the respective cargo spaces, along with the transfer of empty cargo space E during operation cycles. The correctness of the solution is verified by simulating the operation process.

5.3. Comparison of objective functions

This section aims to establish the superiority between two evaluation metrics: Potential Energy Consumption (PEC) and makespan. To achieve this, we selected one metric as the objective function and compared the optimization results obtained under the other metric to assess the reciprocal influence of the optimization process. For each group of experiments, we collected the following data: (1) PEC and makespan values for the optimized solution when minimizing PEC. (2) PEC and makespan values for the optimized solution when minimizing makespan.

The experimental results are presented in Table 4. For the instance with $T \times M = 50 \times 5$, when minimizing PEC, optimization reaches 20.59%, and makespan is improved by 31.92%. Conversely, when minimizing makespan, optimization amounts to 52.41%, and PEC is enhanced by 11.16%. As this comparison underscores the disparity in the optimization difficulties, from a vertical view, this contrast becomes more evident: (1) Optimizing PEC results in a GAP of 45.79% compared to not optimizing it. (2) Optimizing makespan yields a GAP of 39.09% compared to not optimizing it. In conclusion, PEC not only delivers higher optimization benefits compared to makespan, but also offers a more comprehensive approach, encompassing both energy consumption and efficiency. This distinction becomes increasingly pronounced as the task scale grows.

Upon closer examination of the iteration curves, illustrated in Fig. 12, it is evident that with the proposed PEC as the objective function, both the PEC and makespan iteration curves converge, leading to dual optimization. In contrast, when makespan is utilized, the PEC curve exhibits continuous oscillations throughout the iteration process, resulting in pronounced optimization instability. This observation underscores the comprehensive nature of the PEC metric and its superior suitability for the integrated optimization of storage space and crane scheduling.

To assess the sensitivity of the PEC to physical changes in the crane, we conducted a sensitivity analysis of the objective function. We adjusted key physical parameters by ±5% and ±10% and observed the resulting variations in the objective function. The results, shown in Table 5, indicate that the objective function is more sensitive to crane weight and speed, while being less affected by changes in acceleration.

5.4. Comparative results of the number of shuttles

In this final section, We evaluate the optimization results achieved by the algorithms with both metrics, PEC and makespan, for varying the number of shuttles. As shown in Fig. 13, as the number of shuttles increases, the PEC decreases at first and increases when the number of shuttles is excessive. In this particular instance with $N = 60$, the optimal number of shuttles is 3. This is because as the capacity increases, although the crane can fulfill more R/S requests in a single operation cycle, the average load of the crane also increases linearly when leaving the I/O port and during movements, resulting in a larger energy consumption. From this perspective, we can ascertain the most energy efficient number of shuttles for crane.

In contrast, as depicted in Fig. 14, when makespan is employed as the objective function, the curve continuously decreases with an increasing number of shuttles, without reaching an optimal point. This is due to the reduction in crane trips to the I/O port for SKUs exchange,

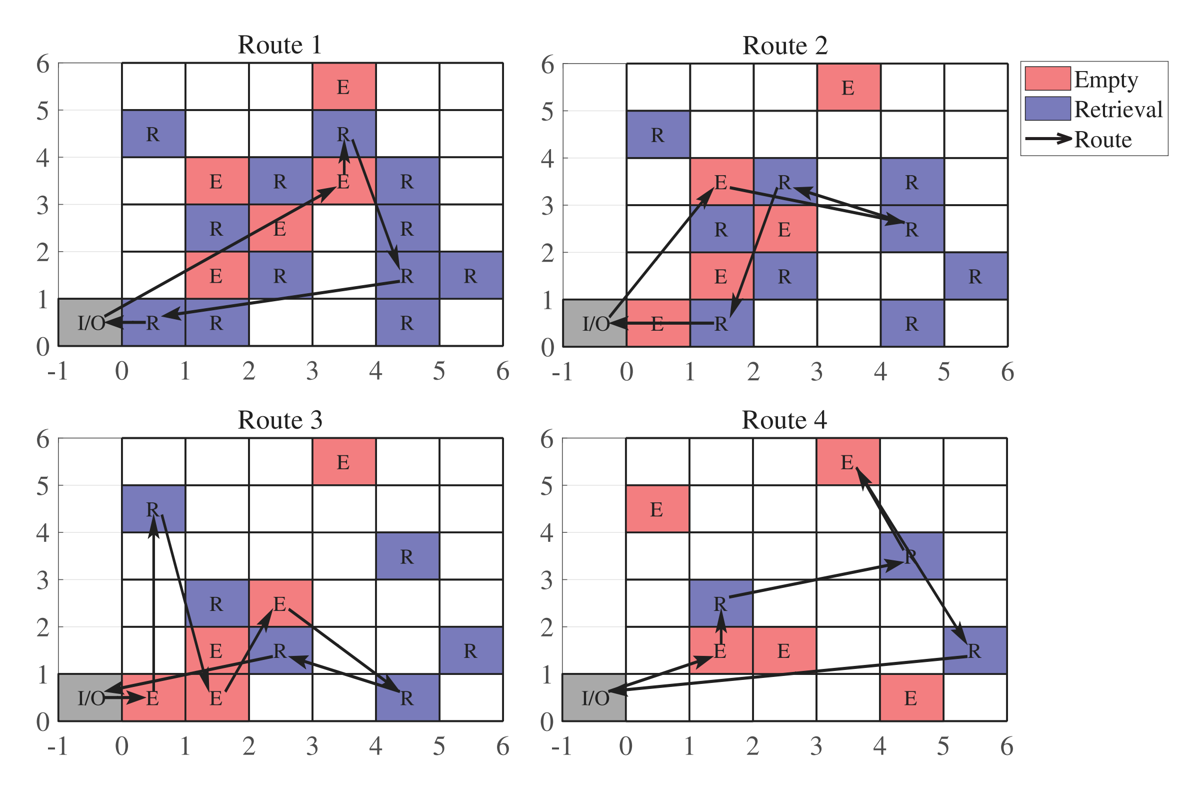


Fig. 11. Route map of the crane in all duty cycles. (For visualization purposes, we provide a GIF animation that illustrates the storage and retrieval process, corresponding to a scenario where $N \times M \times T = 150 \times 5 \times 30$. The animation can be accessed through this link: https://drive.google.com/file/d/1D-adj87GRxJ9sDDSjovXT7bX0ND1rW9M/view?usp=drive_link).

Table 3
Analysis of numerical experimental results.

Group	Average GAP of fitness (%)				Variance of the fitness				Average Runtime (s)			
	RE	GA	HGA	HGA-CC	RE	GA	HGA	HGA-CC	RE	GA	HGA	HGA-CC
1	36.93	19.02	16.24	0.00	0.35	0.86	0.41	0.72	10.72	23.89	25.09	23.32
2	37.21	18.68	16.90	0.00	1.03	1.32	0.73	0.48	12.27	38.14	37.68	36.48
3	34.98	18.46	15.07	0.00	1.01	1.74	0.81	1.98	18.92	93.42	90.74	87.37
4	35.02	17.67	13.11	0.00	0.50	2.40	0.63	1.49	42.82	150.77	147.53	143.40
5	32.62	16.08	14.03	0.00	1.57	5.05	4.66	3.04	44.32	170.95	166.68	159.90
6	32.14	16.35	14.26	0.00	4.04	5.49	4.22	1.52	77.41	433.37	428.64	412.02
7	32.00	17.36	14.74	0.00	0.83	5.77	3.90	4.85	89.40	617.77	614.70	580.17
8	31.83	16.56	13.89	0.00	7.88	2.83	1.15	3.42	89.96	693.54	691.27	661.40
Mean	34.09	17.52	14.78	0.00	2.15	3.18	2.06	2.19	48.23	277.73	275.29	263.01

Table 4
Comparative experimental results of two evaluation indicators.

Percentage decrease of different indicators compared to the beginning											
N × M × T		30 × 3 × 10		50 × 5 × 10		100 × 4 × 25		120 × 4 × 30		150 × 5 × 30	
Indicator		PEC	MS	PEC	MS	PEC	MS	PEC	MS	PEC	MS
Selection of objective function	PEC	0.2062	0.2867	0.2059	0.3192	0.2177	0.2521	0.1897	0.2217	0.1765	0.2126
	makespan	0.1247	0.4858	0.1116	0.5241	0.1128	0.452	0.0674	0.3631	0.0757	0.3791
Gap (%)		39.525	40.984	45.799	39.096	48.186	44.226	64.470	38.942	57.110	43.920

leading to a decrease in the overall travel distance. However, this conclusion appears overly idealistic, as the number of shuttles cannot be expanded infinitely due to physical space limitations.

Overall, it is advisable to avoid overdesigning the crane capacity during the initial construction of the AS/RS, as it can effectively lower energy costs and enhance economic efficiency.

6. Conclusion

In this paper, we delve into the integrated optimization of storage allocation and crane scheduling in automated storage and retrieval systems. This integrated optimization comprises three components: the assignment of R/S requests, the allocation of storage spaces, and the route planning of multi-shuttle crane.

The inclusion of shared storage as an optional component introduces variability in the number of crane movements across operation cycles. To address this challenge, we propose a Multi-layer Adaptive Length Coding (MALC) method, effectively mapping the solution space to the problem space. The Adaptability of coding length ensures highly scalability for the problem. We formulate this integrated optimization with a Mixed Integer Programming (MIP), A novel evaluation metric called Potential Energy Consumption (PEC) is introduced as the objective function. This metric considers the weight and throughput frequency of storage requests, influencing the selection of SKUs

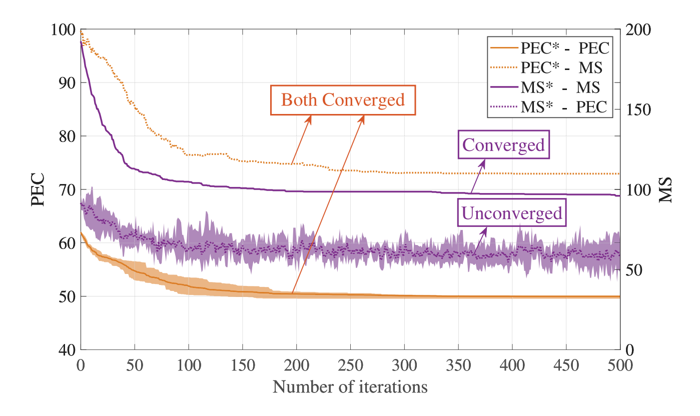


Fig. 12. Interaction of each evaluation indicator during iteration (The objective function is marked with an asterisk).

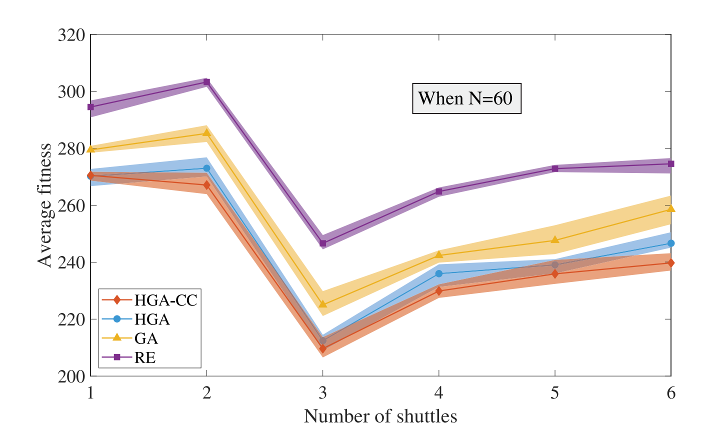


Fig. 13. Average fitness values of algorithms for different number of shuttles with PEC as the objective function.

Table 5
The sensitivity analysis of the objective function.

The range of parameters variation	The sensitivity of the objective function to parameters change		
	Horizontal & Vertical speed	Acceleration	Crane weight
10%	9.78%	1.86%	9.45%
5%	4.90%	0.97%	4.72%
0	0	0	0
−5%	−4.92%	−1.08%	−4.72%
−10%	−9.83%	−2.27%	−9.44%

Table 6
The average fitness values of HGA-CC for various shuttle quantities, utilizing the two evaluation metrics as objective functions, respectively.

N		1	2	3	4	5	6
M = 60	PEC	270.5	267.1	209.6	229.9	235.9	239.8
	MS	1305	926.8	826.6	744.8	721.2	688.2

for specific storage spaces on the shelves. To tackle this problem, a Hybrid Genetic Algorithm with Cooperative Coevolution (HGA-CC) is proposed. The algorithm comprises a global search and a local search, decomposing the problem into multiple subproblems. It demonstrates impressive efficiency in handling closely similar subproblems encountered in optimizing the crane's single-operation cycle.

Numerical experiments were conducted to validate the optimality and stability of the proposed HGA-CC in comparison with other algorithms as depicted in Table 6. The results also demonstrates that PEC as an objective function encompasses the makespan of the crane operation cycles simultaneously. This leads to the both-convergence

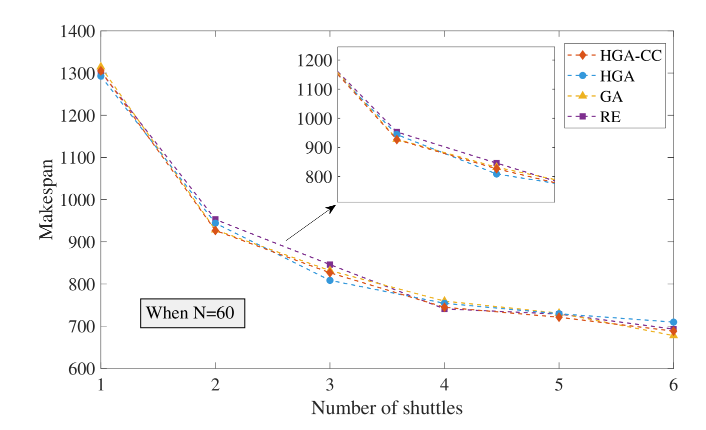


Fig. 14. Average fitness values of algorithms for different number of shuttles with makespan as the objective function.

of the iterative curves, achieving a state of dual-optimization that is unattainable when using makespan. Hence the new evaluation metric proves to be more suitable for the integrated optimization of storage space allocation and crane scheduling in automated storage and retrieval systems. Moreover, these flesh evaluation metrics provide valuable insights for the physical construction of such systems [33].

However, this research still has some limitations. Further exploration is needed to devise solutions for scenarios with an uneven number of storage and retrieval requests [34]. For instance, some systems may encounter over 90% of retrieval requests in the morning at the start of operations, and over 90% of storage requests in the evening at closing. Addressing these specific cases constitutes a crucial next step in our research.

CRediT authorship contribution statement

Wenbin Zhang: Writing – original draft, Visualization, Software, Methodology, Conceptualization. **Zhiyun Deng:** Validation, Investigation, Conceptualization. **Chunjiang Zhang:** Validation, Methodology, Investigation. **Weiming Shen:** Writing – review & editing, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The work presented in this paper has been partially funded by the Key Research and Development Program of the Ministry of Science and Technology of China under Grant 2022YFE0114200, and by the National Natural Science Foundation of China under Grant U20A6004.

Data availability

Data will be made available on request.

References

[1] Y. Yu, Y. Liu, H. Yu, Optimal two-class-based storage policy in an AS/RS with two depots at opposite ends of the aisle, Int. J. Prod. Res. 60 (15) (2022) 4668–4692, http://dx.doi.org/10.1080/00207543.2021.1934590.

[2] B. Bigliardi, G. Casella, E. Bottani, Industry 4.0 in the logistics field: A bibliometric analysis, IET Collab. Intell. Manuf. 3 (1) (2021) 4–12, http://dx.doi.org/10.1049/cim2.12015.

[3] J.-P. Gagliardi, J. Renaud, A. Ruiz, Models for automated storage and retrieval systems: a literature review, Int. J. Prod. Res. 50 (24) (2012) 7110–7125, http://dx.doi.org/10.1080/00207543.2011.633234.
[4] R.D. Meller, A. Mungwattana, Multi-shuttle automated storage/retrieval systems, IIE Trans. 29 (10) (1997) 925–938, http://dx.doi.org/10.1023/A:1018592017528.
[5] Y. Li, Z. Li, Shuttle-based storage and retrieval system: A literature review, Sustainability 14 (21) (2022) 14347, http://dx.doi.org/10.3390/su142114347.
[6] N. Boysen, K. Stephan, A survey on single crane scheduling in automated storage/retrieval systems, European J. Oper. Res. 254 (3) (2016) 691–704, http://dx.doi.org/10.1016/j.ejor.2016.04.008.
[7] L. Polten, S. Emde, Multi-shuttle crane scheduling in automated storage and retrieval systems, European J. Oper. Res. 302 (3) (2022) 892–908, http://dx.doi.org/10.1016/j.ejor.2022.01.043.
[8] L. Ghomri, Z. Sari, Mathematical modeling of the average retrieval time for flow-rack automated storage and retrieval systems, J. Manuf. Syst. 44 (2017) 165–178, http://dx.doi.org/10.1016/j.jmsy.2017.05.002.
[9] P. Yang, L. Miao, Z. Xue, B. Ye, Variable neighborhood search heuristic for storage location assignment and storage/retrieval scheduling under shared storage in multi-shuttle automated storage/retrieval systems, Transp. Res. E 79 (2015) 164–177, http://dx.doi.org/10.1016/j.tre.2015.04.009.
[10] H.H. Tostani, H. Haleh, S.H. Molana, F. Sobhani, A Bi-Level Bi-Objective optimization model for the integrated storage classes and dual shuttle cranes scheduling in AS/RS with energy consumption, workload balance and time windows, J. Clean. Prod. 257 (2020) 120409, http://dx.doi.org/10.1016/j.jclepro.2020.120409.
[11] F. Xiao, Z.-H. Hu, K.-X. Wang, P.-H. Fu, Spatial distribution of energy consumption and carbon emission of regional logistics, Sustainability 7 (7) (2015) 9140–9159, http://dx.doi.org/10.3390/su7079140.
[12] B. Salah, O. Janeh, B. Noche, T. Bruckmann, S. Darmoul, Design and simulation based validation of the control architecture of a stacker crane based on an innovative wire-driven robot, Robot. Comput.-Integr. Manuf. 44 (2017) 117–128, http://dx.doi.org/10.1016/j.rcim.2016.08.010.
[13] D. Zhang, L. Pee, L. Cui, Artificial intelligence in E-commerce fulfillment: A case study of resource orchestration at Alibaba's smart warehouse, Int. J. Inf. Manage. 57 (2021) 102304, http://dx.doi.org/10.1016/j.ijinfomgt.2020.102304.
[14] A. Kattepur, Workflow composition and analysis in industry 4.0 warehouse automation, IET Collab. Intell. Manuf. 1 (3) (2019) 78–89, http://dx.doi.org/10.1049/iet-cim.2019.0017.
[15] K. Azadeh, R. De Koster, D. Roy, Robotized and automated warehouse systems: Review and recent developments, Transp. Sci. 53 (4) (2019) 917–945, http://dx.doi.org/10.1287/trsc.2018.0873.
[16] F.J. Aldarondo, Y.A. Bozer, Expected distances and alternative design configurations for automated guided vehicle-based order picking systems, Int. J. Prod. Res. 60 (4) (2022) 1298–1315, http://dx.doi.org/10.1080/00207543.2020.1856438.
[17] C. Ding, H. He, W. Wang, W. Yang, Y. Zheng, Optimal strategy for intelligent rail guided vehicle dynamic scheduling, Comput. Electr. Eng. 87 (2020) 106750, http://dx.doi.org/10.1016/j.compeleceng.2020.106750.
[18] T.-S. Lee, S.-J. Huang, S.-H. Dai, Contactless power transfer for rail-guided vehicles with power equalization and efficiency improvement considerations, IEEE Trans. Ind. Electron. 69 (4) (2021) 3566–3576, http://dx.doi.org/10.1109/TIE.2021.3076728.
[19] P. Yang, L. Miao, Z. Xue, L. Qin, An integrated optimization of location assignment and storage/retrieval scheduling in multi-shuttle automated storage/retrieval systems, J. Intell. Manuf. 26 (2015) 1145–1159, http://dx.doi.org/10.1007/s10845-013-0846-7.
[20] P. Yang, Z. Zhao, H. Guo, Order batch picking optimization under different storage scenarios for e-commerce warehouses, Transp. Res. E 136 (2020) 101897, http://dx.doi.org/10.1016/j.tre.2020.101897.
[21] K.J. Roodbergen, I.F. Vis, A survey of literature on automated storage and retrieval systems, European J. Oper. Res. 194 (2) (2009) 343–362, http://dx.doi.org/10.1016/j.ejor.2008.01.038.
[22] Y.A. Bozer, J.A. White, Travel-time models for automated storage/retrieval systems, IIE Trans. 16 (4) (1984) 329–338, http://dx.doi.org/10.1080/07408178408975252.
[23] R.L. Francis, L.F. McGinnis, J.A. White, Facility Layout and Location : An Analytical Approach, second ed., Prentice Hall, 1992.
[24] Z. Sui, T. Zhang, T. Wu, H. Chen, Storage optimization of three-dimensional warehouse based on multi population space mapping genetic algorithm, J. Jilin Univ.(Science Edition) 60 (1) (2022) 127–134, http://dx.doi.org/10.13413/j.cnki.jdxblxb.2021037.
[25] W. Zhao, T. Lian, L. Zhang, Y. Zhang, M. Ran, X. Liu, Optimization method for storage location assignment in automated warehouse based on digital twin, Aeronaut. Manuf. Technol. 66 (6) (2023) 66–73, http://dx.doi.org/10.1016/j.micpro.2020.103356.
[26] W.H. Hausman, L.B. Schwarz, S.C. Graves, Optimal storage assignment in automatic warehousing systems, Manage. Sci. 22 (6) (1976) 629–638, http://dx.doi.org/10.1287/mnsc.22.6.629.
[27] J. Li, M. Moghaddam, S.Y. Nof, Dynamic storage assignment with product affinity and ABC classification—a case study, Int. J. Adv. Manuf. Technol. 84 (2016) 2179–2194, http://dx.doi.org/10.1007/s00170-015-7806-7.
[28] X. Yan, Z. Zhang, Q. Liu, C. Lv, L. Zhang, S. Li, An NSABC algorithm for multi-aisle AS/RS scheduling optimization, Comput. Ind. Eng. 156 (2021) 107254, http://dx.doi.org/10.1016/j.cie.2021.107254.
[29] N. Boysen, S. Emde, K. Stephan, Crane scheduling for end-of-aisle picking: Complexity and efficient solutions based on the vehicle routing problem, EURO J. Transp. Logist. 11 (2022) 100085, http://dx.doi.org/10.1016/j.ejtl.2022.100085.
[30] T. Zhang, Y. Li, X. Cao, Motor power calculation and choosing of stacker travelling mechanism, Equip. Manuf. Technol. (1) (2011) 61–63, http://dx.doi.org/10.3969/j.issn.1672-545X.2011.01.023.
[31] Z. Deng, J. Fan, Y. Shi, W. Shen, A coevolutionary algorithm for cooperative platoon formation of connected and automated vehicles, IEEE Trans. Veh. Technol. 71 (12) (2022) 12461–12474, http://dx.doi.org/10.1109/TVT.2022.3196366.
[32] J. Liu, S. Feng, Q. Niu, L. Ji, New construction heuristic algorithm for solving the vehicle routing problem with time windows, IET Collab. Intell. Manuf. 1 (2019) 90–96, http://dx.doi.org/10.1049/iet-cim.2019.0035.
[33] S. Derhami, J.S. Smith, K.R. Gue, A simulation-based optimization approach to design optimal layouts for block stacking warehouses, Int. J. Prod. Econ. 223 (2020) 107525, http://dx.doi.org/10.1016/j.ijpe.2019.107525.
[34] H. Liu, Q. Chen, N. Pan, Y. Sun, Y. An, D. Pan, UAV stocktaking task-planning for industrial warehouses based on the improved hybrid differential evolution algorithm, IEEE Trans. Ind. Inform. 18 (1) (2021) 582–591, http://dx.doi.org/10.1109/TII.2021.3054172.